Package 'phers'

June 1, 2022

Type Package

Title Calculate Phenotype Risk Scores

Version 0.0.3

Description Use phenotype risk scores based on linked clinical and genetic data to study Mendelian disease and rare genetic variants. See Bastarache et al. 2018 <doi:10.1126/science.aal4043>.

URL https://phers.hugheylab.org, https://github.com/hugheylab/phers

License GPL-2

Encoding UTF-8

LazyData true

RoxygenNote 7.1.2

Depends R (>= 3.5)

Imports checkmate (>= 2.0.0), data.table (>= 1.5.0), foreach (>= 1.5.2), iterators (>= 1.0.14), BEDMatrix (>= 2.0.3)

Suggests doParallel (>= 1.0.17), knitr, rmarkdown, testthat (>= 3.1.0), qs (>= 0.25.2)

Config/testthat/edition 3

NeedsCompilation no

Author Jake Hughey [aut, cre], Layla Aref [aut]

Maintainer Jake Hughey <jakejhughey@gmail.com>

Repository CRAN

Date/Publication 2022-05-31 22:20:05 UTC

R topics documented:

`demoSample` *Sample table of demographic information*

Description

The data are artificial and do not correspond to real patients.

Usage

```
demoSample
```

Format

A data table with the following columns:

- `person_id`: Character vector of the identifier for each person in the cohort
- `sex`: Character vector indicating biological sex

See Also

`getWeights()`, `getScores()`

diseaseDxIcdMap *Mapping of diseases and diagnostic ICD codes*

Description

This table provides a mapping between 27 Mendelian diseases and the corresponding ICD-9 and ICD-10 codes that indicate a genetic diagnosis.

Usage

```
diseaseDxIcdMap
```

Format

A data.table with the following columns:

- `disease_id`: Numeric vector of OMIM disease identifiers
- `disease_name`: Character vector of disease names
- `icd`: Character vector of ICD codes indicating a genetic diagnosis
- `flag`: Integer vector of the vocabulary of the ICD code (**9**: ICD-9-CM, **10**: ICD-10-CM)
- `icd_name`: Character vector containing the description of each ICD code

See Also

`getPhecodeOccurrences()`, `getDxStatus()`

diseaseHpoMap *Mapping of Mendelian diseases and their clinical features*

Description

This table provides a mapping between Mendelian diseases and their clinical features, represented as Human Phenotype Ontology (HPO) terms. The mapping is based on annotations from Online Mendelian Inheritance in Man (OMIM).

Usage

```
diseaseHpoMap
```

Format

A data.table with the following columns:

- `disease_id`: Numeric vector of OMIM disease identifiers
- `disease_name`: Character vector of disease names
- `hpo_term_id`: Character vector of HPO identifiers corresponding to each disease's clinical features
- `hpo_term_name`: Character vector of HPO term descriptions

Source

https://hpo.jax.org/app/download/annotation

See Also

mapDiseaseToPhecode()

---

getDxStatus *Identify cases and controls for Mendelian diseases*

Description

This function is useful for verifying that raw or residual phenotype risk scores of diagnosed individuals (cases) tend to be higher than scores of undiagnosed individuals (controls).

Usage

```
getDxStatus(
  demos,
  icdOccurrences,
  minUniqueDates = 2L,
  diseaseDxIcdMap = phers::diseaseDxIcdMap
)
```

Arguments

demos	A data.table having one row per person in the cohort. Must have a column person_id.
icdOccurrences	A data.table of occurrences of ICD codes for each person in the cohort. Must have columns person_id, icd, flag, and entry_date.
minUniqueDates	Integer indicating the minimum number of unique ICD code entry dates required to classify a person as a case. Persons with at least one, but fewer than minUniqueDates entry dates, are assigned as neither cases nor controls.
diseaseDxIcdMap	A data.table of the mapping between diseases and the corresponding ICD codes that indicate a diagnosis. Must have columns disease_id, icd, and flag. Default is diseaseDxIcdMap.

Value

A data.table with columns person_id, disease_id, and dx_status (1 indicates a case, 0 indicates a control, -1 indicates neither).

See Also

phers()

Examples

```
library('data.table')

dxStatus = getDxStatus(demoSample, icdSample)
```

`getGeneticAssociations` *Perform association tests between phenotype risk scores and genotypes*

Description

The association test for each disease-variant pair is based on a linear model, with the phenotype risk score as the dependent variable.

Usage

```
getGeneticAssociations(
  scores,
  genotypes,
  demos,
  diseaseVariantMap,
  lmFormula,
  modelType = c("genotypic", "additive", "dominant", "recessive"),
  level = 0.95,
  dopar = FALSE
)
```

Arguments

`scores`	A data.table of phenotype risk scores. Must have columns `person_id`, `disease_id`, `score`.
`genotypes`	A matrix or 'BEDMatrix' object containing genetic data, with rownames corresponding to `person_id`s in `demos` and `scores`, and colnames corresponding to `variant_id`s in `diseaseVariantMap`.
`demos`	A data.table of characteristics for each person in the cohort. Must have column `person_id`.
`diseaseVariantMap`	A data.table indicating which genetic variants to test for association with phenotype risk scores for which diseases. Must have columns `disease_id` and `variant_id`.
`lmFormula`	A formula representing the linear model (excluding the term for genotype) to use for the association tests. All terms in the formula must correspond to columns in `demos`.
`modelType`	A string indicating how to encode genotype in the model.

level	A number indicating the level of the confidence interval. Default is 0.95.
dopar	Logical indicating whether to run calculations in parallel if a parallel backend is already set up, e.g., using doParallel::registerDoParallel(). Recommended to minimize runtime.

Value

A data.table of statistics for the association tests (if a model fails to converge, NAs will be reported):

- disease_id: Disease identifier
- variant_id: Variant identifier
- n_total: Number of persons with non-missing genotype data for the given variant.
- n_wt: Number of persons homozygous for the wild-type allele.
- n_het: Number of persons having one copy of the alternate allele.
- n_hom: Number of persons homozygous for the alternate allele.
- beta: Coefficient for the association of genotype with score
- se: Standard error for beta
- pval: P-value for beta being non-zero
- ci_lower: Lower bound of the confidence interval for beta
- ci_upper: Upper bound of the confidence interval for beta

If modelType is "genotypic", the data.table will include separate statistics for heterozygous and homozygous genotypes.

See Also

stats::lm(), stats::confint(), getScores(), phers()

Examples

```
library('data.table')
library('BEDMatrix')

# map ICD codes to phecodes
phecodeOccurrences = getPhecodeOccurrences(icdSample)

# calculate weights
weights = getWeights(demoSample, phecodeOccurrences)

# OMIM disease IDs for which to calculate phenotype risk scores
diseaseId = 154700

# map diseases to phecodes
diseasePhecodeMap = mapDiseaseToPhecode()

# calculate scores
scores = getScores(
  demoSample, phecodeOccurrences, weights, diseasePhecodeMap[disease_id == diseaseId])
```

```
# map diseases to genetic variants
nvar = 10
diseaseVariantMap = data.table(disease_id = diseaseId, variant_id = paste0('snp', 1:nvar))

# load sample genetic data
npop = 50
genoSample = BEDMatrix(system.file('extdata', 'geno_sample.bed', package = 'phers'))
colnames(genoSample) = paste0('snp', 1:nvar)
rownames(genoSample) = 1:npop

# run genetic association tests
genoStats = getGeneticAssociations(
  scores, genoSample, demoSample, diseaseVariantMap, lmFormula = ~ sex,
  modelType = 'additive')
```

`getPhecodeOccurrences` *Map ICD code occurrences to phecode occurrences*

Description

This is typically the first step of an analysis using phenotype risk scores, the next is `getWeights()`.

Usage

```
getPhecodeOccurrences(
  icdOccurrences,
  icdPhecodeMap = phers::icdPhecodeMap,
  dxIcd = phers::diseaseDxIcdMap
)
```

Arguments

`icdOccurrences`	A data.table of occurrences of ICD codes for each person in the cohort. Must have columns `person_id`, `icd`, and `flag`.
`icdPhecodeMap`	A data.table of the mapping between ICD codes and phecodes. Must have columns `icd`, `phecode`, and `flag`. Default is the map included in this package.
`dxIcd`	A data.table of ICD codes to exclude from mapping to phecodes. Must have columns `icd` and `flag`. Default is the table of Mendelian diseases and the corresponding ICD codes that indicate a genetic diagnosis. If `NULL`, no ICD codes will be excluded.

Value

A data.table of phecode occurrences for each person.

See Also

`getWeights()`, `getScores()`, `phers()`

Examples

```
library('data.table')

# map ICD codes to phecodes
phecodeOccurrences = getPhecodeOccurrences(icdSample)

# calculate weights
weights = getWeights(demoSample, phecodeOccurrences)

# OMIM disease IDs for which to calculate phenotype risk scores
diseaseId = 154700

# map diseases to phecodes
diseasePhecodeMap = mapDiseaseToPhecode()

# calculate scores
scores = getScores(
  demoSample, phecodeOccurrences, weights, diseasePhecodeMap[disease_id == diseaseId])

# calculate residual scores
rscores = getResidualScores(demoSample, scores, lmFormula = ~ sex)
```

---

getResidualScores *Calculate residual phenotype risk scores*

Description

The residual score indicates to what extent a person's phenotype risk score for a given disease deviates from the expected score, after adjusting for the person's characteristics in a linear model.

Usage

```
getResidualScores(demos, scores, lmFormula)
```

Arguments

demos	A data.table of characteristics for each person in the cohort. Must have column person_id.
scores	A data.table containing the phenotype risk score for each person for each disease. Must have columns person_id, disease_id, and score.
lmFormula	A formula representing the linear model to use for calculating residual scores. All terms in the formula must correspond to columns in demos.

Value

A data.table, based on scores, with an additional column resid_score. Residual scores for each disease are standardized to have unit variance.

See Also

stats::rstandard(), getScores(), phers()

Examples

```
library('data.table')

# map ICD codes to phecodes
phecodeOccurrences = getPhecodeOccurrences(icdSample)

# calculate weights
weights = getWeights(demoSample, phecodeOccurrences)

# OMIM disease IDs for which to calculate phenotype risk scores
diseaseId = 154700

# map diseases to phecodes
diseasePhecodeMap = mapDiseaseToPhecode()

# calculate scores
scores = getScores(
  demoSample, phecodeOccurrences, weights, diseasePhecodeMap[disease_id == diseaseId])

# calculate residual scores
rscores = getResidualScores(demoSample, scores, lmFormula = ~ sex)
```

---

getScores *Calculate phenotype risk scores*

Description

A person's phenotype risk score for a given disease corresponds to the sum of the weights of the disease-relevant phecodes that the person has received.

Usage

```
getScores(demos, phecodeOccurrences, weights, diseasePhecodeMap)
```

Arguments

demos	A data.table having one row per person in the cohort. Must have a column person_id.
phecodeOccurrences	A data.table of phecode occurrences for each person in the cohort. Must have columns person_id and phecode.
weights	A data.table of phecodes and their corresponding weights. Must have columns phecode and w.
diseasePhecodeMap	A data.table of the mapping between diseases and phecodes. Must have columns disease_id and phecode.

Value

A data.table containing the phenotype risk score for each person for each disease.

See Also

mapDiseaseToPhecode(), getPhecodeOccurrences(), getWeights(), getResidualScores(), phers()

Examples

```
library('data.table')

# map ICD codes to phecodes
phecodeOccurrences = getPhecodeOccurrences(icdSample)

# calculate weights
weights = getWeights(demoSample, phecodeOccurrences)

# OMIM disease IDs for which to calculate phenotype risk scores
diseaseId = 154700

# map diseases to phecodes
diseasePhecodeMap = mapDiseaseToPhecode()

# calculate scores
scores = getScores(
  demoSample, phecodeOccurrences, weights, diseasePhecodeMap[disease_id == diseaseId])

# calculate residual scores
rscores = getResidualScores(demoSample, scores, lmFormula = ~ sex)
```

---

getWeights *Calculate phecode-specific weights for phenotype risk scores*

Description

This is typically the second step of an analysis using phenotype risk scores, the next is getScores().

Usage

```
getWeights(demos, phecodeOccurrences)
```

Arguments

demos	A data.table having one row per person in the cohort. Must have a column person_id.
phecodeOccurrences	A data.table of phecode occurrences for each person in the cohort. Must have columns person_id and phecode.

Value

A data.table with columns phecode, prev (prevalence), and w (weight). Prevalence corresponds to fraction of the cohort that has at least one occurrence of the given phecode. Weight is calculated as -log10 prevalence.

See Also

getPhecodeOccurrences(), getScores(), phers()

Examples

```
library('data.table')

# map ICD codes to phecodes
phecodeOccurrences = getPhecodeOccurrences(icdSample)

# calculate weights
weights = getWeights(demoSample, phecodeOccurrences)

# OMIM disease IDs for which to calculate phenotype risk scores
diseaseId = 154700

# map diseases to phecodes
diseasePhecodeMap = mapDiseaseToPhecode()

# calculate scores
scores = getScores(
  demoSample, phecodeOccurrences, weights, diseasePhecodeMap[disease_id == diseaseId])

# calculate residual scores
rscores = getResidualScores(demoSample, scores, lmFormula = ~ sex)
```

---

hpoPhecodeMap *Mapping of HPO terms and phecodes*

Description

This table provides a mapping between Human Phenotype Ontology (HPO) terms and phecodes, and is useful for using phecodes to represent the clinical features of Mendelian diseases.

Usage

```
hpoPhecodeMap
```

Format

A data.table with the following columns:

- `hpo_term_id`: Character vector of HPO term identifiers
- `hpo_term_name`: Character vector of HPO term descriptions
- `phecode`: Character vector of phecodes
- `phecode_name`: Character vector of phecode descriptions

See Also

`mapDiseaseToPhecode()`

`icdPhecodeMap` *Mapping of ICD codes and phecodes*

Description

This table provides a mapping between International Classification of Diseases 9th and 10th revisions (ICD-9-CM and ICD-10-CM) and phecodes (version 1.2).

Usage

```
icdPhecodeMap
```

Format

A data.table with the following columns:

- `icd`: Character vector of ICD codes
- `flag`: Integer vector of the vocabulary of the ICD code (**9**: ICD-9-CM, **10**: ICD-10-CM)
- `icd_name`: Character vector of ICD code descriptions
- `phecode`: Character vector of phecodes
- `phecode_name`: Character vector of phecode descriptions

Source

https://phewascatalog.org/phecodes

See Also

`getPhecodeOccurrences()`

icdSample *Sample table of ICD occurrences*

Description

The data are artificial and do not correspond to real patients.

Usage

```
icdSample
```

Format

A data.table with the following columns:

- `person_id`: Character vector of the identifier for each person
- `icd`: Character vector of the ICD codes recorded for each person
- `flag`: Integer vector of the vocabulary of the ICD code (**9**: ICD-9-CM, **10**: ICD-10-CM)
- `entry_date`: Vector of type `Date` indicating the date each ICD code was recorded.

See Also

`getPhecodeOccurrences()`, `getWeights()`, `getScores()`

mapDiseaseToPhecode *Map diseases to phecodes via HPO terms*

Description

A mapping of diseases to their clinical features, represented as phecodes, is required for calculating phenotype risk scores.

Usage

```
mapDiseaseToPhecode(
  diseaseHpoMap = phers::diseaseHpoMap,
  hpoPhecodeMap = phers::hpoPhecodeMap
)
```

Arguments

`diseaseHpoMap`	A data.table containing the mapping between diseases and HPO terms. Must have columns `disease_id` and `term_id`. Default is the map included in this package.
`hpoPhecodeMap`	A data.table containing the mapping between HPO terms and phecodes. Must have columns `term_id` and `phecode`. Default is the map included in this package.

Value

A data.table with columns `disease_id` and `phecode`.

See Also

`getScores()`, `phers()`

Examples

```
library('data.table')

# map ICD codes to phecodes
phecodeOccurrences = getPhecodeOccurrences(icdSample)

# calculate weights
weights = getWeights(demoSample, phecodeOccurrences)

# OMIM disease IDs for which to calculate phenotype risk scores
diseaseId = 154700

# map diseases to phecodes
diseasePhecodeMap = mapDiseaseToPhecode()

# calculate scores
scores = getScores(
  demoSample, phecodeOccurrences, weights, diseasePhecodeMap[disease_id == diseaseId])

# calculate residual scores
rscores = getResidualScores(demoSample, scores, lmFormula = ~ sex)
```

`phers` *Perform multiple steps of an analysis using phenotype risk scores*

Description

This function can map ICD occurrences to phecode occurrences, calculate weights for each phecode, and calculate raw and residual phenotype risk scores.

Usage

```
phers(
  demos,
  icdOccurrences,
  diseasePhecodeMap,
  icdPhecodeMap = phers::icdPhecodeMap,
  dxIcd = phers::diseaseDxIcdMap,
  weights = NULL,
  residScoreFormula = NULL
)
```

Arguments

demos	A data.table having one row per person in the cohort. Must have a column person_id.
icdOccurrences	A data.table of occurrences of ICD codes for each person in the cohort. Must have columns person_id, icd, and flag.
diseasePhecodeMap	A data.table of the mapping between diseases and phecodes. Must have columns disease_id and phecode.
icdPhecodeMap	A data.table of the mapping between ICD codes and phecodes. Must have columns icd, phecode, and flag. Default is the map included in this package.
dxIcd	A data.table of ICD codes to exclude from mapping to phecodes. Must have columns icd and flag. Default is the table of Mendelian diseases and the corresponding ICD codes that indicate a genetic diagnosis. If NULL, no ICD codes will be excluded.
weights	A data.table of phecodes and their corresponding weights. Must have columns phecode and w. If NULL (the default), weights will be calculated based on data for the cohort provided. If the cohort is small or its phecode prevalences do not reflect those in the population of interest, it is recommended to use preCalcWeights.
residScoreFormula	A formula representing the linear model to use for calculating residual scores. All terms in the formula must correspond to columns in demos. If NULL, no residual scores will be calculated.

Value

A list with elements:

- phecodeOccurences: A data.table of phecode occurrences for each person in the cohort.
- weights: A data.table of phecodes and their corresponding weights.
- scores: A data.table of raw and possibly residual phenotype risk scores for each person and each disease.

See Also

getPhecodeOccurrences(), getWeights(), getScores(), getResidualScores(), mapDiseaseToPhecode(), icdPhecodeMap, diseaseDxIcdMap, preCalcWeights, getDxStatus()

Examples

```
library('data.table')

# OMIM disease IDs for which to calculate phenotype risk scores
diseaseId = 154700

# map diseases to phecodes
diseasePhecodeMap = mapDiseaseToPhecode()
```

```
diseasePhecodeMap = diseasePhecodeMap[disease_id == diseaseId]

# calculate raw and residal scores using weights based on the sample cohort
scores = phers(
  demoSample, icdSample, diseasePhecodeMap, residScoreFormula = ~ sex)

# calculate scores using pre-calculated weights
scores = phers(
  demoSample, icdSample, diseasePhecodeMap,
  weights = phers::preCalcWeights, residScoreFormula = ~ sex)
```

`preCalcWeights` *Pre-calculated weights for calculating phenotype risk scores*

Description

The weights are based on EHR data from the Vanderbilt University Medical Center Synthetic Derivative (SD) and ICD-phecode map version 1.2.

Usage

```
preCalcWeights
```

Format

A data.table with the following columns:

- `phecode`: Character vector of phecodes
- `prev`: Numeric vector of prevalences, i.e., fraction of subjects in the SD that have at least one occurrence of the given phecode
- `w`: Numeric vector of weights, calculated as `-log10(prev)`

See Also

`getWeights()`

Index